

# Prueba Técnica: Desarrollo de una Aplicación Móvil y Backend

## Objetivo:

Desarrollar una aplicación móvil para gestionar una lista de tareas colaborativa con un backend que provea las APIs necesarias. La aplicación debe permitir a múltiples usuarios colaborar en tiempo real y persistir los datos.

---

## Requisitos Técnicos

### 1. Frontend (React Native con Expo):

- Crear una aplicación móvil que permita:
  - Registrar un nuevo usuario (autenticación básica).
  - Ver una lista de tareas compartida.
  - Agregar, editar y eliminar tareas.
  - Actualizar en tiempo real la lista de tareas cuando un usuario haga un cambio.
- **Características mínimas:**
  - Diseño limpio y responsivo.
  - Uso de Context API o un manejador de estado (Redux/Zustand).
  - Manejo de errores (por ejemplo, conexión fallida al backend).
  - Implementar una pantalla de inicio de sesión y registro.
  - Soporte para temas claro/oscuro (opcional, pero deseado).

### 2. Backend (NestJS):

- Proveer APIs RESTful para:
  - Registro e inicio de sesión de usuarios (utilizando JWT para autenticación).
  - Operaciones CRUD para las tareas.
  - Sincronización en tiempo real (cualquier estrategia, a consideración propia)
- **Características mínimas:**
  - Implementación modular y limpia siguiendo principios de diseño como SOLID.
  - Validación de datos con `class-validator` o similares) y manejo de excepciones.
  - Uso de una base de datos como PostgreSQL (usar Prisma).
  - Configuración del backend para que sea fácilmente ejecutable en un entorno local (por ejemplo, con Docker).

### 3. General:

- Documentar el proyecto en un archivo `README.md`, incluyendo:
  - Cómo configurar y ejecutar el frontend y el backend.
  - Descripción de las funcionalidades implementadas.
  - Explicación de cualquier decisión técnica clave.

---

## Tareas a Completar

### 1. Funcionalidades del Frontend:

- Implementar un flujo de autenticación básica (registro e inicio de sesión).
- Mostrar una lista de tareas colaborativa (cargar tareas desde el backend).
- Permitir agregar, editar y eliminar tareas desde la app.
- Actualizar automáticamente la lista de tareas para todos los usuarios conectados.

### 2. Funcionalidades del Backend:

- Crear APIs seguras para manejar usuarios y tareas.
  - Implementar sincronización en tiempo real.
  - Asegurarse de que solo usuarios autenticados puedan realizar operaciones.
- 

## Criterios de Evaluación

### 1. Calidad del Código:

- Estructura modular y reutilizable.
- Uso adecuado de buenas prácticas (por ejemplo, separación de lógica y presentación en React Native, módulos y servicios en NestJS).

### 2. Cumplimiento de Requisitos:

- Completar las funcionalidades solicitadas.
- Correcto manejo de autenticación y sincronización en tiempo real.

### 3. Documentación:

- Claridad y detalle en el [README .md](#).

### 4. Creatividad:

- Uso de librerías adicionales para mejorar la experiencia de usuario.

### 5. Pruebas:

- Inclusión de pruebas unitarias y/o de integración, especialmente en el backend.
- 

## Extras (Opcionales):

- Implementar pruebas automáticas en el backend y/o frontend.
  - Crear un flujo para compartir listas de tareas con otros usuarios mediante enlaces.
  - Implementación de arquitecturas limpias.
  - Implementación de características opcionales, como el tema claro/oscuro
-

## Entregables:

1. Repositorio en GitHub con dos carpetas:
  - `/frontend` para la app de React Native.
  - `/backend` para la API de NestJS.
2. Instrucciones detalladas en el `README.md` para ejecutar ambos proyectos.
3. Código limpio, documentado y funcional.
4. Tiempo máximo de la prueba es de 48 hr, hacer un primer commit al inicio de proyecto para tomar hora de inicio.
5. En caso de necesitar más tiempo, notificarlo con 24hr de antelación.